EXHIBIT 6

DOCKET NO: 0100157-00243

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

PATENT: 5,978,791

INVENTOR: DAVID A. FARBER
AND RONALD D. LACHMAN

FILED: OCT. 24, 1997 ISSUED: NOV. 2, 1999

TITLE: DATA PROCESSING SYS-
TEM USING SUBSTANTIALLY
UNIQUE IDENTIFIERS TO IDEN-
TIFY DATA ITEMS, WHEREBY
IDENTICAL DATA ITEMS HAVE
THE SAME IDENTIFIERS

Mail Stop PATENT BOARD
Patent Trial and Appeal Board
U.S. Patent & Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 5,978,791 UNDER 35 U.S.C. § 312 AND 37 C.F.R. § 42.104

TABLE OF CONTENTS

TABLE OF AUTHORITIES

Page(s)

I. MANDATORY NOTICES

A. Real Parties-in-Interest

EMC Corporation and VMware, Inc. ("Petitioner") are the real parties-in-interest.

B. Related Matters

The '791 patent is the first issued of an extensive patent family of continuation and divisional applications. Exhibit 1008 shows the patent family, with patents in red and blue including the '791 patent being asserted in the litigation *PersonalWeb Technologies LLC v. EMC Corporation and VMware, Inc.* (No. 6:11-cv-00660-LED) (E.D. Tex.), served on December 16, 2012.

Petitioner is also seeking *inter partes* review of related U.S. Patents Nos. 6,415,280, 7,945,539, 7,945,544, 7,949,662, and 8,001,096, and requests that they be assigned to the same Board for administrative efficiency.

C. Counsel

Lead Counsel: Peter M. Dichiara (Registration No. 38,005)

Backup Counsel: David L. Cavanaugh (Registration No. 36,476)

D. Service Information

Email: Peter Dichiara, peter.dichiara@wilmerhale.com

Post and Hand Delivery: WilmerHale, 60 State St., Boston MA 02109

Telephone: 617-526-6466 Facsimile: 617-526-5000

E. Certification of Grounds for Standing

Petitioner certifies pursuant to Rule 42.104(a) that the patent for which review is sought is available for *inter partes* review and that Petitioner is not barred or estopped from requesting an *inter partes* review challenging the patent claims on the grounds identified in this Petition.

II. OVERVIEW OF CHALLENGE AND RELIEF REQUESTED

A. Prior Art Patents and Printed Publications

Pursuant to Rules 42.22(a)(1) and 42.104 (b)(1)-(2), Petitioner challenges claims 1-4, 29-33 and 41 of U.S. Patent No. 5,978,791 ("the '791 patent", Ex. 1001) as anticipated by or unpatentable in view of the following patents and printed publications:

1. S. Browne et al., "Location-Independent Naming for Virtual Distributed Software Repositories," University of Tennessee Technical Report CS-95-278 (Feb. 1995) ("Browne ", Ex. 1002).[1]

---

[1]The Browne February 1995 publication qualifies as prior art under 35 USC 102(a), and is used in this petition because it includes illustrations facilitating explanation of the invalidity of the challenged claims. Petitioner also has attached as exhibits and included in its claim charts two earlier versions of this publication – S. Browne et al., "Location-Independent Naming for Virtual Distributed Software Repositories," http://www.netlib.org/utk/papers/lifn/main.html (Nov. 11, 1994) (Exhibit 1006); and K. Moore et al., "An Architecture for Bulk File Distribution,"

2. Albert Langer, "Re: dl/describe (File descriptions)," post to the "alt.sources" newsgroup on August 7, 1991 ("Langer", Ex. 1003)[2]
3. Kantor, "The Frederick W. Kantor Contents-Signature System Version 1.22," FWKCS122.REF (August 10, 1993) ("Kantor", Ex. 1004).[3]
4. Woodhill et al., U.S. Patent No. 5,649,196, entitled "System and Method For Distributed Storage Management on Networked Computer Systems Using Binary Object Identifiers," filed Nov. 9, 1995 as a continuation of application 85,596, filed July 1, 1993 ("Woodhill", Ex. 1005).

Network Working Group Internet Draft (July 27, 1994) (Ex. 1007). As Dr. Clark confirms in his Decl., the references are substantially the same with respect to the relevant disclosures. If the Patent Owner alleges an earlier priority date of the challenged claims, Petitioner may rely on the earlier publications for invalidity, alone or in combination with the other references cited in this petition .

[2] Langer was made available on the "alt.sources.d" and "comp.archives.admin" newsgroup distribution lists on August 7, 1991. Both newsgroups were widely disseminated and readily accessible to the relevant technical community.

[3] Kantor's FWKCS manual has been publicly and freely available continuously since August,1993. Kantor distributed the user manual with the FWKCS program as shareware and posted it online to electronic Bulletin Board Systems including "The Invention Factory" and "Channel 1" for an extended period of time, where it could be downloaded by anyone. (*See* Kantor at 3; *see also* 158-59; Ex. 1004.)

B. There is a Reasonable Likelihood that at least One Claim of the '791 Patent is Unpatentable Under 35 U.S.C. §§ 102, 103

Section VI below explains how the above-cited patents and printed publications create a reasonable likelihood that Petitioner will prevail with at least one of the challenged claims. *See* 35 U.S.C. § 314(a). Indeed, that section supported by the attached claim charts of Exhibits 1038-1041 and the Declaration of Dr. Douglas Clark, a Professor of Computer Science at Princeton University ("Clark Decl."; Ex. 1009), demonstrate that all of the challenged claims are anticipated by, or unpatentable in view of, each of these references.

C. Relief Requested

Petitioner requests cancellation of claims 1-4, 29-33, and 41, the challenged claims, as unpatentable under 35 U.S.C. §§ 102 and 103.

III. Claim Construction

The claim terms should be given their "broadest reasonable construction in light of the specification." 37 C.F.R. § 42.100(b). In prior litigation involving the '791 patent a court construed two claim terms of the '791 patent:

Claim Term	Construction
"substantially unique identifier"	"an identity for a data item generated by processing *all* of the data in the data item, and *only* the data in the data item, through an algorithm"
"using the identifier"	"employing the unique identifier of the data item, with or without other information, to carry out the recited function"

(Order Regarding Claim Construction, Akamai Tech., Inc, v. Digital Island, Inc., No. 00-11851-RWZ (D. Mass. Nov. 8, 2011), ECF No. 137; Ex. 1037.)

The claim terms can be understood by their ordinary and plain meaning except where construed in the specification. The specification includes the following constructions relevant to the challenged claims:

Term	**Construction**
"data" and "data item"	"as used herein refer to sequences of bits. Thus a data item may be the contents of a file, a portion of a file, a page in memory, an object in an object-oriented program, a digital message, a digital scanned image, a part of a video or audio signal, or any other entity which can be represented by a sequence of bits. ('791 patent, col. 1, ll. 54-60, s*ee also* col. 1, l. 65 – col. 2, l. 2 ("data items (the data items being files, directories, records in the database, objects in an object-oriented programming, locations in memory or on a physical device or the like)"); Ex. 1001.)
"location"	"with respect to a data processing system, refers to any of a particular processor in the system, a memory of a particular processor, a storage device, a removable storage medium (such as a floppy disk or compact disk), or any other physical location in the system" ('791 patent, col. 5, l. 65- col. 6, l. 4; Ex. 1001.)
"True Name, data iden-	"refer to the substantially unique data identifier for a particular item" ('791 patent, col. 6, ll. 7-10, s*ee also*

Term	Construction
tity, and data identifier"	col. 14, ll. 1-39 (describing mechanism for calculating True Name using MD hash function); Ex. 1001.)

Certain challenged claims of the '791 patent include limitations which are recited in means-plus-function form and should be interpreted under 35 U.S.C. § 112 ¶ 6. These limitations are identified below in conjunction with the structure, material or acts described in the '791 patent corresponding to the recited function.[4]

Claim Term	Function and Corresponding Structure
"identity means for determining, for any of a plurality of data items present in the system, a substantially unique identifier, the identifier being determined using and depending on all of the data in the data item and only the data in the data item, whereby two identical data items in the system will have the same	Function: Determining, for any of a plurality of data items present in the system, a substantially unique identifier, the identifier being determined using and depending on all of the data in the data item and only the data in the data item, whereby two identical data items in the system will have the same identifier. Structure: The processor of Fig. 1(b), programmed to execute the "Calculate True Name" primitive mechanism depicted in Figures 10(a) and 10(b), where the MD function is one of the MD4, MD5 and SHA functions. *See* '791 patent at 4:64–6:19, 12:54–14:39, 14:51–53, 31:32–50, 32:54–64 (Ex. 1001); U.S. Patent No. 6,415,280 Prosecution History, Response (Aug. 22, 2001) at 27 (Ex. 1019).

[4] The corresponding structure may be insufficient to satisfy 35 USC §112, ¶¶2, 6.

Claim Term	Function and Corresponding Structure
identifier" (claim 1)	
"existence means for determining whether a particular data item is present in the system, by examining the identifiers of the plurality of data items" (claim 1)	Function: Determining whether a particular data item is present in the system, by examining the identifiers of the plurality of data items. Structure: The processor of Fig. 1(b), storing the True File Registry and programmed to execute the "Locate Remote File" primitive mechanism depicted in Figures 16(a) and 16(b). *See* '791 patent at 4:64–6:19, 9:36–10:10, 16:38–17:9, 17:41–43, 25:10–13, 35:51–55 (Ex. 1001).
"local existence means for determining whether an instance of a particular data item is present at a particular location in the system, based on the identifier of the data item" (claim 2)	Function: Determining whether an instance of a particular data item is present at a particular location in the system, based on the identifier of the data item. Structure : The processor of Fig. 1(b), storing the True File Registry and programmed to execute the "Locate True File" remote mechanism depicted in Figure 28. *See* '791 patent at 4:64–6:19, 9:36–10:10, 16:48–51, 23:52–24:28, 32:42–45, 35:51–55, 36:65–66 (Ex. 1001); U.S. Patent No. 6,415,280 Prosecution History, Response (Aug. 22, 2001) at 28 (Ex. 1019).
"local existence means for determining whether an instance of a particular data item is present at a particu-	Function: Determining whether a particular data item is present at a particular location in the system by examining the identifiers of the plurality of data items at said particular location in the system. Structure: The processor of Fig. 1(b), storing the True

Claim Term	Function and Corresponding Structure
lar location in the system, based on the identifier of the data item" (claim 3)	File Registry and programmed to execute the "Locate True File" remote mechanism depicted in Figure 28. *See* '791 patent at 4:64–6:19, 9:36–10:10, 16:48–51, 23:52–24:28, 32:42–45, 35:51–55, 36:65–66 (Ex. 1001).
"data associating means for making and maintaining, for a data item in the system, an association between the data item and the identifier of the data item" (claim 4)	Function: Making and maintaining, for a data item in the system, an association between the data item and the identifier of the data item. Structure: The processor of Fig. 1(b), storing the True File Registry and programmed to execute the "Assimilate Data Item" primitive mechanism depicted in Figure 11. *See* '791 patent at 4:64–6:19, 9:36–10:10, 14:40–15:4, 15:41–44, 16:29–31, 18:34–36, 18:43–45, 19:30–37, 24:34–35, 24:51–52, 28:30–33, 30:55–57, 32:54–33:9, 33:33–39 (Ex. 1001); U.S. Patent No. 6,415,280 Prosecution History, Response (Aug. 22, 2001) at 39 (Ex. 1019).
"access means for accessing a particular data item using the identifier of the data item" (claim 4)	Function: Accessing a particular data item using the identifier of the data item. Structure: The processor of Fig. 1(b), storing the True File Registry and programmed to execute the "Make True File Local" primitive mechanism depicted in Figures 17(a) and 17(b). *See* '791 patent at 4:64–6:19, 9:36–10:10, 17:10–45, 18:4–8, 18:53–55, 20:65–21:5, 24:54–55, 33:51–59, 36:52–55, 36:61–64 (Ex. 1001).

IV. OVERVIEW OF THE '791 PATENT

A. Brief Description

The '791 patent is directed to data storage systems that use "substantially unique [data] identifiers" – based on all the data in a data item and only the data in the data item – to identify and access data items. (*See, e.g.*, '791 patent, Title, Abstract, and col. 1, ll. 13-18; Ex. 1001.) The patent uses these identifiers to perform basic file management functions such as eliminating unnecessary duplicate copies of computer files or other data items—an admittedly old problem. (See, e.g., '791 patent, Background of the Invention, col. 2, ll. 46-57; Ex. 1001.)

According to the patent, prior art systems identified data items based on their location or address within the data processing system. ('791 patent, col. 1, ll. 23-28; Ex. 1001.) For example, files were often identified by their context or "pathname," that is, information specifying a path through the computer directories to the particular file (e.g., C:\My Documents\Law School\1L\TortsOutline.txt). ('791 patent, col. 1, ll. 35-42; Ex. 1001.) The patent contends that all prior art systems operated in this manner: "In ***all*** of the prior data processing systems, the names or identifiers provided to identify data items. . . are ***always*** defined relative to a specific context," and "there is ***no*** direct relationship between the data names and the data item." ('791 patent, col. 1, l. 65 – col. 2. l. 3, col. 2, ll. 12-13, emphasis added; Ex. 1001.)

According to the patent, this prior art practice of identifying a data item by its context or pathname resulted in certain shortcomings. For example, with pathname identification, the same data name may refer to different data items, or conversely, two different data names may refer to the same data item. ('791 patent, col. 2, ll. 12-16; Ex. 1001.) Moreover, because there is no correlation between the contents of a data item and its pathname, there is no *a priori* way to confirm that the data item is in fact the one named by the pathname. ('791 patent, col. 2, ll. 18-21; Ex. 1001.) Furthermore, context or pathname identification may more easily result in the creation of unwanted duplicate data items, *e.g.*, multiple copies of a file on a file server.[5] ('791 patent, col. 2, ll. 47-58; Ex. 1001.)

The '791 patent purports to address these shortcomings. ('791 patent, col. 3, ll. 6-20; Ex. 1001.) It suggests that "it is therefore desirable to have a mechanism . . . to determine a common and substantially unique identifier for a data item, using only the data in the data item and not relying on any sort of context." ('791 patent, col. 3., ll. 6-11; Ex. 1001.) Moreover, "[i]t is further desirable to have a mechanism for reducing multiple copies of data items… and to have a mechanism which

[5] For example, Alice and Bob both download the same copy of the James Bond movie *Goldfinger*. Alice saves her copy at "C:\Movies\Bond\Goldfinger.mov", and Bob saves his copy at "C:\Videos\007\Bond-Goldfinger.mov".

enables the identification of identical data items so as to reduce multiple copies." ('791 patent, col. 3, ll. 12-15; Ex. 1001.)

To do so, the '791 patent provides data identifiers that "depend[] on all of the data in the data item and only on the data in the data item." ('791 patent, col. 1, ll. 13-18; *see also* col. 3, ll. 29-32; Ex. 1001.) Preferred embodiments use either of the well-known MD5 or SHA message digest functions[6] to calculate a substantially unique identifier from the contents of the data item. ('791 patent, col. 12, l. 54-col. 14, l. 39; Ex. 1001.) The system first computes the 16-byte (128-bit) message digest of the data item and then appends the size of the data item to produce a 160-bit identifier. ('791 patent, Fig. 10A and col. 14, ll. 1-12; Ex. 1001.) The patent calls these context- or location-independent, content-based identifiers a "True Name" – a phrase admittedly "coined by the inventors." (U.S. Patent No. 6,415,280 Prosecution History, Response (Aug. 22, 2001), at 22; Ex. 1019.)

---

[6] A message digest function or hash is a transformation of a piece of data into a much shorter form. (*See, e.g.*, D. Banisar et al., The Third CSPR Cryptography and Privacy Conference at 509 (p. 14 of PDF) (1993) (describing a message digest function as "a 128-bit cryptographically strong one-way hash function of the message" that is "somewhat analogous to a 'checksum' or CRC error checking code, in that it compactly 'represents' the message."); Ex. 1010.)

With these identifiers, the patent asserts, "data items can be accessed by reference to their identities (True Names) independent of their present location." ('791 patent, col. 34, ll. 9-11; *see also* col. 34, ll. 30-32; Ex. 1001.) The actual data item corresponding to these location-independent identifiers may reside anywhere, e.g., locally, remotely, offline. ('791 patent, col. 34, ll. 11-19; Ex. 1001.) "Thus the identity of a data item is independent of its name, origin, location, address, or other information not derivable directly from the data, and depends only on the data itself." ('791 patent, col. 3, ll. 33-35; Ex. 1001.)

In the preferred embodiments, the substantially unique identifiers are used to "augment" standard file management functions of an existing operating system. ('791 patent, col. 6, ll. 11-19; Ex. 1001.) For example, a local directory extensions (LDE) table[7] is indexed by a pathname or contextual name of a file and also includes True Names for most files. ('791 patent, col. 8, ll. 19-26; Ex. 1001.) A True File registry (TFR) lists True Names, and stores "location, dependency, and migration information about True Files." ('791 patent, col. 8, ll. 27-28, 33-35; Ex. 1001.) True Files are identified in the True File registry by their True Names, and can be looked up in the registry by their True Names. ('791 patent, col. 8, ll. 30–

[7] The patent describes an LDE table as a data structure which provides information about files and directories in the system and includes information in addition to that provided by the native file system. ('791 patent, col. 8, ll. 19-26; Ex. 1001.)

32; col. 23, ll. 61–62; Ex. 1001.) This look-up provides, for each True Name, a list of the locations, such as file servers, where the corresponding file is stored. ('791 patent, col. 34, ll. 17–19; *see also* col. 16, ll. 11–13; Ex. 1001.)

When a data item is to be "assimilated" into the data processing system, its substantially unique identifier (True Name) is calculated and compared to the True File Registry to see if the True Name already exists in the Registry. ('791 patent, col. 14, ll. 41-56; Ex. 1001.) If the True Name already exists, this means that the data item already exists in the system and the to-be-assimilated data item (i.e., the scratch file) need not be stored. ('791 patent, col. 14, ll. 56-60; Ex. 1001.) Conversely, if the True Name does not exist in the Registry, then a new entry is created in the Registry which is then set to the just-calculated True Name value, and the data items can be stored. ('791 patent, col. 14, ll. 61-67; Ex. 1001.)

The "True Name of a file can be used to identify a file by contents, to confirm that a file matches its original contents, or to compare two files." ('791 patent, col. 15, ll. 25-27; Ex. 1001.) The patent asserts that "[w]hen a data item is to be copied to another location… all that is necessary is to examine the True Name of the data item prior to the copying. If a data item with the same True Name already exists at a destination location locally … then there is no need to copy the data item." ('791 patent, col. 33, ll. 28-33; Ex. 1001.)

B. The Prosecution History of the '791 Patent

The '791 patent is based on an application that was originally filed on April 11, 1995. Initial claim 1 of the application read as follows:

> 1. In a data processing system, an apparatus comprising:
>
> identity means for determining, for any of a plurality of data items in the system, a substantially unique identifier, said identifier depending on all of the data in the data item and only on the data in the data item; and
>
> existence means for determining whether a particular data item is present in the system, by examining the identifiers of the plurality of data items.

(Application as Filed on April 11, 1995, at 77; Ex. 1024.) All claims were rejected as anticipated by Gramlich et al. (U.S. Pat. No. 5,202,982) or as being unpatentable over Gramlich in view of Konrad et al. (U.S. Pat. No.5,404,508). (Office Action of September 7, 1996, Ex. 1025.)

In response, applicants amended the claims and emphasized that their substantially unique identifiers were based on "all" and "only" the data in the data items. (Amendment of March 12, 1997 at 10-11, emphasis in original; Ex. 1026.) The claims were again rejected as anticipated by, or unpatentable in view of, Gramlich and other prior art. (Office Action of May 30, 1997; Ex. 1027.)

The applicants amended the claims a second time, arguing that their invention required substantially unique identifiers based on "all" and "only" the data in the data items:

> This invention relates to data processing systems and, more particularly, to data processing systems wherein data items are identified by substantially unique identifiers which:
>
> (A) depend on and
>
> (B) are determined using:
>
> (a) **all** of the data in the data items and
>
> (b) **only** the data in the data items.
>
> A notable and significant property of this invention is that, in any particular system, two identical data items in the system will have the same identifier.

(Amendment of August 29, 1997 at 8, 10, emphasis in the original, Ex. 1028.)

Claim 1 was eventually issued after a file wrapper continuation application and some other procedural issues were addressed.

V. THE CHALLENGED CLAIMS ARE UNPATENTABLE

A. There is Nothing New About using Identifiers that Depend On All and Only the data of the data item

The '791 claims focus on the concept of using a "substantially unique identifier" – based on "all" and "only" the data in a data item – to perform basic file management functions. Claims 1 and 30 of the patent, for example, require simply (i) determining the identifier, and (ii) using the identifier to determine if the data item is present in the system or to access the data item:

> 1. In a data processing system, an apparatus comprising:
>
> identity means for determining, for any of a plurality of data items present in the system, a substantially unique identifier, the identifier being determined using and depending on all of the data in the data

item and only the data in the data item, whereby two identical data items in the system will have the same identifier; and

existence means for determining whether a particular data item is present in the system, by examining the identifiers of the plurality of data items.

30. A method of identifying a data item present in a data processing system for subsequent access to the data item, the method comprising:

determining a substantially unique identifier for the data item, the identifier depending on and being determined using all of the data in the data item and only the data in the data item, whereby two identical data items in the system will have the same identifier; and

accessing a data item in the system using the identifier of the data item.

('791 patent, col. 39, ll. 14-24, col. 42, ll. 58-67; Ex. 1001.)

The applicants indicated in their patent application that they were entitled to these broad claims because "[i]n ***all*** of the prior data processing systems, the names or identifiers provided to identify data items . . . are ***always*** defined relative to a specific context," and "there is ***no direct relationship*** between the data names and the data item." ('791 patent, col. 1, l. 65–col. 2. l. 3, col. 2, ll. 12-13, emphasis added; Ex. 1001.) They further argued to the PTO that the '791 approach was inventive because it used data identifiers based on "all" and "only" the data in a data item. (Amend. of March 12, 1997 at 10–11; Ex. 1026.)

These representations were simply wrong. Prior data processing system ***did use*** identifiers based on the data in a data item itself, and not its context or path-name. In fact, these techniques were old and widely used. This is not surprising. The concept of using a mathematical function to create a "fingerprint" or "signature" for a data item based on the content of the data item predates the '791 patent by decades. For example, IBM developed one of the first hash tables in the 1950s (*see*, *e.g.*, D. Knott, Hashing functions, The Computer Journal 18 (1975), vol. 3, at 273-74 (discussing "history of hashing"); Ex. 1011), and Professor Ron Rivest of MIT introduced the MD5 algorithm referenced in the '791 patent in the early 1990s. (*See, e.g.*, R. Rivest, "The MD5 Message-Digest Algorithm," Internet RFC 1321 (Apr. 1992); Ex. 1012.) These hashing functions take as input the data contained in a file, or other data item, and produce a much smaller-sized output value, commonly called a "hash," "hash value," "message digest" ("MD"), or "checksum." (*See, e.g.*, McGraw-Hill Dictionary of Scientific and Technical Terms, (4th ed., 1989), at 860; Ex. 1013; *see also* B. Kaliski, "A Survey of Encryption Standards," IEEE Micro (Dec. 1993), pp. 74–81, at 77; Ex. 1014.) For example, a file that is a million bytes (or even much larger) in size can be used as input to produce a hash value that is a mere 16 bytes in length. Because of the mathematical properties of the function, the odds that two different files will produce the identical 16 byte hash are extremely small: for example, with a 16 byte hash output, the

odds that two randomly picked inputs have the same hash are 2^{-64}, or approximately one in sixteen billion billions. (B. Kaliski at 77; Ex. 1014.)

Consequently, hashes are known as "signatures" or "fingerprints" because they identify data with high reliability, just like signatures or fingerprints are used to identify people with a high degree of certainty. (*See* D.R. McGregor and J.A. Mariani, 'Fingerprinting' – A Technique for File Identification and Maintenance, Software Practice & Experience 1165 (1982), vol. 12, no. 12, at 1165 ("fingerprinting" technique "produce[s] a quasi-unique identifier for a file, derived from that file's contents. . .[t]he idea is to provide an identifying feature for every file, which is intrinsically distinctive, and analogous (hopefully) to a human's fingerprint."); Ex. 1017.)

Although applicants suggested in their application that they were the first to utilize these hashing functions to identify data items for file management applications, others working in the field used them for the same purposes more than a decade before the '791 patent. For example, at least sixteen years before the '791 patent was filed, researchers were already using content addressable file stores and a "hash function and bit array" to determine whether two records were identical, and to eliminate duplicate records. (*See*, *e.g.*, Babb, Implementing a Relational Database by Means of Specialized Hardware, ACM Transactions on Database Systems, Vol. 4, No.1, at 2-4, March 1979; Ex. 1029; Bitton and DeWitt at 256 ; Ex.

1030.) Likewise, file "fingerprinting" has long been known as a technique to see if two files were identical. (*See* Rabin, Fingerprinting by Random Polynomials, Center for Research in Computing Technology, Harvard University, Report TR-15-81 at 1 and 9, 1981; Ex. 1015; *see also* Manber, at 3 (commenting on work of Rabin); Ex. 1016). Likewise, the use of "fingerprints" both to identify files and check for duplicates has also long been known. (S*ee, e.g.*, D.R. McGregor and J.A. Mariani, at 1165; Ex. 1017.)[8]

Many printed publications and patents disclose and use data identifiers exactly like those described and claimed in the '791 patent. These publications disclose identifiers that are location- and context-independent, that are determined using all of, and only, the contents of the data item, and that are formed using identical algorithms to those mentioned in the '791 patent.

---

[8] This reference was cited and central to the analysis and rejection of EP counterpart application EP0826181A1 with similar claims. (Annex to the communication dated May 8, 2009; Ex. 1020.) Applicant amended the claims to emphasize a "licensing" limitation not found in the challenged claim (Reply to communication from the Examining Division dated November 18, 2009 at 4; Ex. 1021), but this too was found unpersuasive and the rejection was maintained by the EPO. (Annex to the communication dated March 14, 2012 at 4; Ex. 1022.) Following this rejection, Applicants withdrew the application from consideration. (Closing of Application dated June 14, 2012; Ex. 1023.)

Browne: For example, researchers at the University of Tennessee and Bell Laboratories disclosed a system that created "location independent file name" (or "LIFN") to identify files on the Internet. (Browne at 3; Ex. 1002.) LIFNs – like the identifiers in the '791 patent – uniquely identified files by their contents, not their locations. (Browne at 3; Ex. 1002; *compare* '791 patent, col. 34, ll. 9-11 (True Names used to identify files "independent of their present location"); Ex. 1001.) LIFN <signatures> were computed as "the ascii form of the MD5 signature of the file" – the same function identified in the '791 patent. (Browne at 6; Ex. 1002; *compare* '791 patent, col. 13, ll. 15-16 (MD5 or SHA used to calculate True Name) ; Ex. 1001.) The only inputs to the MD5 function were the contents of the file, and the MD5 signature was thus based on all of, and only, these contents – again, like the identifiers in the '791 patent. (*See* Browne at 6; Ex. 1002.) LIFN signatures could be used to identify files and to determine if they were present in the system; for example, LIFNs were used to identify and access files at file servers dedicated to storing duplicate copies of files ("cache sites" or "mirror sites"). (Browne at 4; Ex. 1002.)

Langer: Another researcher, Albert Langer, also addressed the same problem as the '791 patent and, like Browne, proposed a virtually identical solution. (Langer; Ex. 1003.) Langer was particularly concerned with sharing content on the Internet prior to the rise of the World Wide Web, through the use popular pro-

tocols such as the File Transfer Protocol (FTP). FTP sites, among other things, could be accessed to provide a listing of available files at the site, and also supported a user's ability to select and download files from the site. (*See, e.g.*, P. Deutsch et al., "How to Use Anonymous FTP," Internet RFC 1635 (May 1994); Ex. 1033.) Langer sought to make FTP sites more useful by improving the accessibility of files at the FTP sites. (Langer at 3; Ex. 1003.) Langer specifically addressed the problem of "uniquely identifying files which may have different names and/or be in different directories on different systems," like the '791 applicants, observing that traditional location-based identifiers do not work well for distributed systems. (Langer at 3; Ex. 1003; *compare* '791 patent, col. 2, ll. 17–28; Ex. 1001.) Langer's solution, like the '791 patent, was to "provide a unique identifier for each file which is independent of location." (Langer at 3; Ex. 1003; *compare* '791 patent, col. 3, ll. 29–35; Ex. 1001.) Specifically, Langer disclosed "defining a unique identifier that does NOT include a particular site identifier," by "using a cryptographic hash function such as MD5," i.e., the identical algorithm used in the '791 patent. (Langer at 4; Ex. 1003; *compare* '791 patent, col. 13, ll. 15–17; Ex. 1001.) Langer's identifier used all of and only the contents of a file as input to the MD5 hash, and thus was based on all of, and only, those contents. Langer proposed incorporating the MD5 hash into file descriptions, which could be used by

then-existing search engines, such as *Archie* and *WAIS*.[9] (Langer at 3–6; Ex. 1003.) A user could determine if a file with a specific MD5 hash existed anywhere on the Internet, and if so, access the file on any of the FTP servers that stored a copy of it. (Langer at 5; Ex. 1003.)

Kantor: Dr. Frederick W. Kantor, a physicist from Columbia University, developed yet another example of a context- and location-independent identifier for the same purposes as the '791 patent. Dr. Kantor described a product called FWKCS that created "contents-signatures" for files based on their content and only their content. (Kantor at Preface 2; Ex. 1004.)[10] FWKCS used these contents-signatures to uniquely identify files on a bulletin board system ("BBS"), an online file system considered a precursor to the World Wide Web. (*Id.*; Ex. 1004) The

---

[9] Early search engines such as Archie and WAIS were specialized databases that mainly indexed FTP sites. Users could run searches against these databases to find files on the Internet. (*See* EARN Staff, "Guide to Network Resource Tools," Internet RFC 1580 (March 1994) at 22–38; Ex. 1034 (describing Archie and WAIS) C. Adie, "Network Access to Multimedia Information," Internet RFC 1614 (May 1994), at 29–34; Ex. 1035 (describing WAIS); G. Kessler et al., "A Primer On Internet and TCP/IP Tools," Internet RFC 1739 (Dec. 1994), at 24–27, 31; Ex. 1036 (describing Archie and WAIS).)

[10] Citations to the Preface of Kantor's FWKCS user manual are labeled "Preface." Otherwise, citations refer to the page numbers of the remainder of the manual.

contents-signatures could be used, for example, to detect and eliminate duplicates (*i.e.,* two files with the same signature), and to determine if a file which the user intended to store to the BBS was already present in the system (to avoid storing an unwanted duplicate). (*Id.* ; Ex. 1004.) The contents-signature was a hash of the contents of the file combined with the size of the content, exactly as is the case for one of the embodiments of the '791 patent. (*Id.* at 8; Ex. 1004; *compare* '791 patent, col. 14, ll. 1-12 and Figure 10A; Ex. 1001). Consequently, just like the '791 patent, the FWKCS contents-signatures depended on all of, and only, the data in the data items.

Woodhill: The Woodhill patent provides still another example of the use of context- and location-independent identifiers for the same purposes as the '791 patent. Woodhill created a distributed storage system that used "Binary Object Identifiers" to identify and access files, and to manage file back-ups, among other functions. (Woodhill; Ex. 1005) As Woodhill explains, a "Binary Object Identifier 74 [of Fig. 3] . . . is a unique identifier for each binary object to be backed up." (Woodhill at col. 4, lines 45-47; Ex. 1005). The Binary Object Identifiers included three fields – a CRC value, a LRC value, and a hash value –each calculated from all of, and only, the contents of the binary object. (Woodhill at col. 8, lines 1- 33; Ex. 1005). As Woodhill emphasized, "[t]he critical feature to be recognized in creating a Binary Object Identifier 74 is that the identifier should be based on the

contents of the binary object so that the Binary Object Identifier 74 changes when the contents of the binary object changes." (Woodhill at col. 8, lines 58-62; Ex. 1005.) Woodhill used these identifiers to identify binary objects that had changed since the most recent backup, so that "only those binary objects associated with the file that have changed must be backed up." (Woodhill at col. 9, lines 7-14; Ex. 1005.) "[D]uplicate binary objects, even if resident on different types of computers in a heterogeneous network, can be recognized from their identical Binary Object Identifiers 74." (Woodhill at col. 8, lines 62-65; Ex. 1005.)

These prior art references provide just a handful of many examples of the use of content-based identifiers to perform basic file management functions. Indeed, the application of hash-based identifiers to these functions was so obvious that at least one commentator not only described the applications as "easy," but also posted these ideas publicly "to impede anyone who might independently have had the idea from patenting it." (Williams, "An algorithm for matching text (possibly original)", posted to the "comp.compression" newsgroup on January 27, 1992; Ex. 1031.) In a later paper, also published before the '791 patent, Williams once again identified the same ideas as the patent, plainly stating that "digest algorithms can be used . . . to generate unique fixed-length identifiers for arbitrary blocks of data in situations where the identifier of identical blocks must be the

same . . .." (R. Williams, "An Introduction to Digest Algorithms," Rocksoft (Nov. 1994), at 13; Ex. 1032.)

In short, other than perhaps coining a new phrase – i.e., True Name – for a very old concept, there is absolutely nothing new disclosed or claimed in the '791 patent concerning the use of location-independent, content-based data identifiers.

VI. SPECIFIC GROUNDS FOR PETITION

Pursuant to Rule 42.104(b)(4)-(5) and Practice Guide Fed. Register Vol. 77, No. 27, page 48764, Petitioners have submitted claim charts in connection with this Petition, (attached as Ex. 1038-1041), from the pending litigation between Petitioner and PersonalWeb Technologies LLC. Those charts set forth Petitioners' position with respect to those references and demonstrate that the challenged claims are anticipated and/or unpatentable in view of each of them. Petitioner also submits herewith the Declaration of Dr. Douglas Clark (Ex. 1009), a Professor of Computer Science at Princeton University. Dr. Clark confirms that the charts identify representative subject matter in each reference that teaches every limitation of the challenged claims. He likewise confirms how each claim is anticipated or, at a minimum, rendered obvious by the prior art.

A. Grounds of Invalidity for Challenged Claims 1-4, 29-33 and 41 based on Browne as a Primary Reference

Ground 1: Browne Anticipates Challenged Claims 1-4, 29-33 and 41

Browne was not cited to the USPTO and not considered by the examiner during prosecution of the '791 patent. It is prior art under at least 35 U.S.C. § 102(a) and anticipates each of claims 1–4, 29–33 and 41 of the '791 patent. [11]

Browne describes the Bulk File Distribution ("BFD") package developed by researchers at the University of Tennessee and Bell Laboratories as part of an effort to make scientific software easily accessible over the Internet. (Browne at 1, 6; Ex. 1002.) The BFD package is based on the concept of a "virtual repository," which is a distributed network of physical software repositories, each residing on a different file server. (Browne at 1–2; Ex. 1002.)

Like the '791 patent, Browne begins by discussing the shortcomings of context- or location-dependent file identifiers. At the time, a virtual repository could be implemented using a Uniform Resource Locator (URL) to identify each file.

---

[11] As indicated *supra*, note 1, Petitioners may rely on earlier versions of this article (Ex. 1006 and Ex. 1007), alone or in combination with other references cited in this petition, if the Patent Owner alleges an earlier priority date of the challenged claims,

(Browne at 2; Ex. 1002.) A URL can be used to specify (i) a transfer protocol, (ii) a location, such as a web server, and (iii) a file name. For example, "http://www.netlib.org/index.html" is a URL that identifies a resource to be accessed (i) using the HyperText Transfer Protocol (HTTP), (ii) at an Internet location "www.netlib.org," and (iii) with file name "index.html." (*See, e.g.*, T. Berners-Lee et al., "Uniform Resource Locators (URL)," Internet RFC 1738 (Dec. 1994) ; Ex. 1018.)

The Browne authors identify several problems with the use of location-based identifiers, such as URLs, to access virtual software repositories. Among other things, URLs are inadequate for ensuring the consistency of a software repository. For example, if the content of a file is updated, the corresponding URL may become outdated. (Browne at 2; Ex. 1002.) Moreover, a URL can only identify a single location; if a virtual software repository offers multiple copies of the same file, each copy must be given its own unique URL. (Browne at 2; Ex. 1002.)

In order to address these shortcomings, Browne adopts the same solution that would be later proposed in the '791 patent: associating a unique identifier with the ***contents*** of a file, rather than with the ***location*** of the file. In the BFD package, the identifier is called a "Location Independent File Name," or LIFN. (Browne at 3; Ex. 1002.) Indeed, Browne even refers to its file names as "location independent," the same terminology later used in the '791 patent. (Browne at

3; Ex. 1002; *compare* '791 patent, col. 3, ll. 33–34 ("the identity of a data item is ***independent*** of its name, origin, ***location*** . . . ." .)(emphasis added); Ex. 1001.)

Instead of identifying a physical location, a LIFN uniquely identifies immutable content, i.e., a fixed sequence of bytes. (Browne at 3; Ex. 1002.) As a consequence, two files with identical content will have the same LIFN even if they are stored on two different computers and are given different names.

In Browne's preferred approach, the LIFN depends on ***all*** of the data in a file, and ***only*** on the data in the file. Specifically, the LIFN is computed as the MD5 hash of the contents of the file. (Browne at 6; Ex. 1002.) The MD5 function receives the ***entire*** contents of the file as its input and returns a 128-bit fingerprint: *i.e.*, the sequence of bytes that constitutes the file is the ***only*** input to the MD5 algorithm. (*See* Browne at 6; Ex. 1002.) The general syntax for the LIFN is "lifn:netlib:<signature>", referencing the file access protocol ("lifn," similar to the "http" protocol identifier in a URL), the server handling the request ("netlib"), and the unique MD5 hash used to identify the file[12] ("<signature>"). (Browne at 4, 6;

[12] Dr. Clark confirms that a person of ordinary skill would understand that the file servers rely only on the <signature> to identify a file. (Clark Decl, ¶ 20.) The first part of the LIFN (protocol) identifies a protocol, and the second part (server) identifies the server used for finding the location of a file, but neither identifies the file itself. (Clark Decl., ¶ 20.)

Ex. 1002.) The MD5 algorithm provides a substantially unique fingerprint, meaning that two files with identical content will always have the same MD5 fingerprint, even if they are located on different servers, and even if the server administrators give them different names.

Even the specific way in which the '791 patent deals with "compound data items" tracks the earlier implementation disclosed in Browne. Browne's simple algorithm directly anticipates the flow chart in Figure 10(b) of the '791 patent. (Browne at 2, 5-6; Ex. 1002; *compare* '791 patent, col. 14, ll. 1–27; Ex. 1001.)

Like the '791 patent, Browne also provides means for determining if a file is present on the network, and for accessing the file, based on its identifier. Specifically, a client computer sends a query to a LIFN server including the LIFN <signature> of the desired file to be accessed. (Browne at 4–5; Ex. 1002.) In response, the LIFN server returns a list of file servers that store a copy of the file associated with that LIFN <signature>.[13] (Browne at 4–5; Ex. 1002.) To be clear, this mechanism is just like the True File Registry (TFR) of '791 patent, which receives a True Name and provides a list of file servers that store a copy of that file. ('791 patent, col. 34, ll. 17–19; Ex. 1001.) Once it is determined if the file is present in

[13] Specifically, a LIFN database associates a LIFN with a list of locations corresponding to that LIFN. (Browne at 6.) Since multiple locations may be used to store the file, mirroring of content increases reliability of the system. (*Id.* at 2.)

the system, the client can use the LIFN <signature> to access the file from one of the identified file servers. (Browne at 4–5; Ex. 1002.) Again, the request includes the LIFN <signature> of the desired file.[14] (Browne at 6; Ex. 1002.)

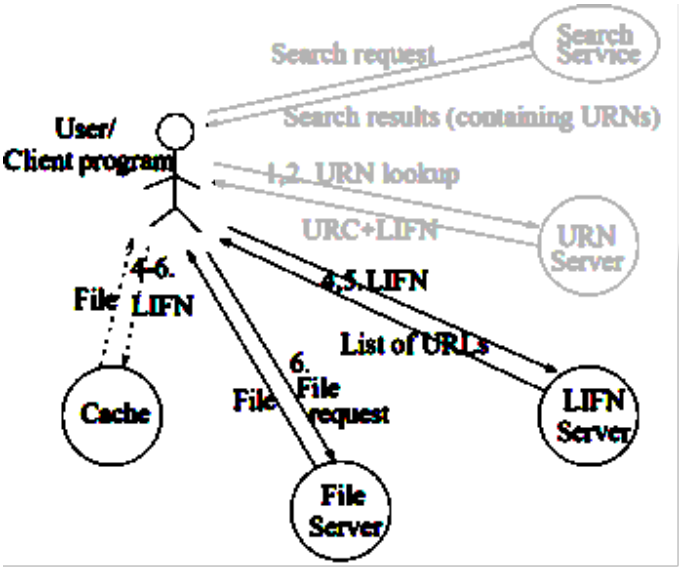
Browne also provides means for determining if a file is locally present on a server, based on the file's substantially unique identifier. The BFD package supports caching.[15] (Browne at 2; Ex. 1002.) In this situation, when a client computer requests a file, it uses the LIFN <signature> to identify the file. The request is received by a cache site, which typically is closer to the user and faster. When the cache site receives the request, it uses the LIFN <signature> to determine if it has a copy of the corresponding file. (Browne at 4; Ex. 1002.) If the cache site has a copy of the file, it can directly provide the file to the client. (Browne at 5; Ex. 1002.) If it does not, the file can be retrieved using the procedure outlined above. (Browne at 5; Ex. 1002.) Browne's caching thus uses a LIFN <signature> as a file identifier to determine if the corresponding file is present in the system (i.e., in the cache) and to access the file.

---

[14] In order to accept a request based on a content-based identifier rather than a location-based file name, a file server "create[s] a directory that aliase[s] the ascii form of the MD5 signatures to the actual file locations." (Browne at 6; Ex. 1002.)

[15] At the time, caching was already a well-known technique in the computer arts.

Figure 3 of Browne (edited for clarity) illustrates the relevant steps in accessing a file.[16] At step 4, the client accesses a LIFN server, as described above, providing a LIFN for the desired file including an MD5 <signature> uniquely identifying the file of interest; at step 5, as described above, the LIFN server provides a list of locations (URLs) for servers that store the file; finally, at step 6, the identifier is used to access the actual file from one of the locations. (Browne at 5; Ex. 1002.) If a cache site is present, steps 4–6 are replaced by a cache lookup, which uses the LIFN signature to determine if it has a copy of the file of interest and, if so, to return it to the user. (Browne at 5; Ex. 1002.)



[16] Steps 1, 2 and 3 (grayed out in the figure) relate to another aspect of the BFD package, the use of Uniform Resource Names or URNs. The use of URNs is not required in order to use LIFNs. (*See* Browne at 4, 6; Ex. 1002.)

As set forth in detail in the attached claim chart (Exhibit 1038),[17] and as confirmed by Dr. Clark (Clark Decl. at ¶ 17-32), Browne anticipates each of claims 1–4, 29–33 and 41 of the '791 patent. For example, claim 1 recites:

> 1. In a data processing system, an apparatus comprising:
>
> identity means for determining, for any of a plurality of data items present in the system, a substantially unique identifier, the identifier being determined using and depending on all of the data in the data item and only the data in the data item, whereby two identical data items in the system will have the same identifier; and
>
> existence means for determining whether a particular data item is present in the system, by examining the identifiers of the plurality of data items.

('791 patent, col. 39, ll. 14–24; Ex. 1001.)

With regard to the "identity means," Dr. Clark explains that Browne meets this element because it calculates a LIFN signature as a "substantially unique identifier" for the file contents. (Clark Decl. at ¶¶19-20; Ex. 1009, Browne at 3, 4, 6 and Figure 2; Ex. 1002.) In doing so, it uses "all of" the data in the file and "only" the data in the file as an input to the MD5 function. (Clark Decl. at ¶ 20; Ex. 1009, Browne at 6; Ex. 1002.) Dr. Clark confirms that Browne's use of MD5 will

[17] Ex. 1038 is from another proceeding and provides information supporting where Browne, Ex. 1002, discloses the limitations of the challenged claims.

result in two identical files always having the same LIFN signature. (Clark Decl. at ¶20; Ex. 1009, Browne at 4, 6; Ex. 1002.) In fact, not only does Browne teach the claim element but it uses the very technique described in the '791 specification. (Clark Decl. at ¶¶ 20, 24 Ex. 1009.) With regard to the "existence means for determining whether a particular data item is present in the system, by examining the identifiers of the plurality of data items," Dr. Clark confirms that Browne teaches this element as well. (Clark Decl. at ¶¶ 21-24; Ex. 1009, Browne at 5, 6 and Figure 3; Ex. 1002.) For example, the BFD system allows a client to query a LIFN server with a LIFN signature to determine a list of locations that is storing the corresponding file. (Clark Decl. at ¶¶ 22-23; Ex. 1009, Browne at 2, 4, 5; Ex. 1002.) In doing so, the LIFN server receives the query and "examines" a list of LIFNs signatures (i.e., "identifiers") to determine a server that holds a file with the requested LIFN signature (i.e., to "determin[e] whether a particular data item [i.e., file] is present in the system"). (Clark Decl. at ¶¶ 22-23; Ex. 1009, Browne at 4; Ex. 1002.) In particular, the LIFN server determines what (if any) locations hold a copy of the file, by examining a LIFN database that stores keys (MD5 values) for each of the files stored on the network. (Clark Decl. at ¶ 22-23; Ex. 1009, Browne at 4, 5; Ex. 1002.) Again, not only does Browne teach the claim element but it uses the very technique described in the '791 specification. (Clark Decl. at ¶ 22-24; Ex. 1009.) In the '791 patent, the True File Registry provides a list of the file

servers that store a file with a given True Name. ('791 patent, col. 34, ll. 17–19; Ex. 1001.)

Browne meets the requirements of Claim 30 – which requires simply determining a substantially unique identifier for a data item and accessing the data item using the identifier – for essentially the same reasons. Dr. Clark identifies the locations of the limitations of the claim in the reference, for example "a method of identifying a data item. . . comprising:" (Clark Decl. at ¶ 22-23; Ex. 1009, Browne at 4, 5 and Figure 3; Ex. 1002.) , "determining a . . . same identifier; and" (Clark Decl. at ¶ 22-23; Ex. 1009, Browne at 3, 4, 6 and Figure 2; Ex. 1002.) and "accessing . . . of the data item" (Clark Decl. at ¶ 22-23; Ex. 1009, Browne at 4, 5 and Figure 3; Ex. 1002.)

A similar analysis reapplies to the remaining claims, as supported by Dr. Clark (Ex. 1009) and the claim chart (Ex. 1038). As Dr. Clark confirms, Browne discloses, for example, a "local existence means" (claim 2 and 3) at p. 5 and Fig. 3; "each location contains . . . at said particular location in the system" (claim 3) at pp. 1-2, 5-6; a "data associating means" and an "access means" (claim 4) at pp. 2-6 and Fig. 3; "at least a file, database. . . object class" (claim 29) at p. 3; "making and maintaining. . . via the association" (claim 31) at p. 5 and Fig. 2-3; "assimilating a new data item. . .with its identifier" (claim 32) at pp. 4-6; "for a given data identifier . . . to the current location" (claim 41) at p. 5 and Fig. 3; and "based on the de-

termining. . . present at the destination" (claim 33) at p. 5. The remaining limitations of claim 33 are similar to those in claim 30, and the discussion, *supra*, is incorporated by reference.

Ground 2: Challenged Claims 1-4, 29-33 and 41 are Unpatentable as Obvious in view of Browne

In the event PersonalWeb contends that Browne does not satisfy the claim limitation of an "identity means" or "the identifier being determined using and depending on . . . only the data in the data item" because Browne's LIFN includes a protocol identifier and a server identifier in addition to the file identifier (i.e., the LIFN <signature>), a person of ordinary skill would have found it obvious to modify Browne to meet that limitation. Dr. Clark confirms that Browne expressly discusses an example where a single LIFN server accepts requests from clients. (Clark Decl. at ¶ 20, 34; Ex. 1009.) In that case, the "netlib:" server identifier and the "lifn:" protocol identifier serve no purpose. (*Id.*) As Dr. Clark further confirms, reducing the LIFN to the MD5 <signature> alone would be desirable in view of the resulting simplification of the system. (*Id.*) The reduction of the LIFN to the MD5 <signature> would constitute the application of a known technique to a known device, ready for improvement, to yield predictable results, and therefore it would be obvious to a person of ordinary skill in the art. (*Id.*)

Ground 3: Challenged Claims 1-4, 29-33 and 41 are Unpatentable as Obvious in view of Browne in combination with Langer

It also would have been obvious to modify Browne to use an identifier consisting only of an MD5 hash by combining it with another reference such as that Langer. Dr. Clark confirms that such a combination would have been desirable to simplify file access where a more sophisticated syntax allowing for multiple protocols and multiple LIFN databases is not required, as in the application expressly discussed at page 6 of Browne. (Clark Decl. at ¶ 35; Ex. 1009; Browne at 6; Ex. 1002.) The modified BFD package would meet the limitation of "the identifier being determined using and depending on all of the data in the data item and only the data in the data item." (*Id.*) The application of Langer's MD5 hash identifier to Browne would constitute the application of a known technique to a known device, ready for improvement, to yield predictable results, and therefore it would have been obvious to a person of ordinary skill in the art. (Clark Decl. at ¶ 36; Ex. 1009, Browne at 1, 2; Ex. 1002, Langer at 3, 4; Ex. 1003.)

Ground 4: Challenged Claims 1-4 and 29 are Unpatentable as Obvious in view of Browne in combination with Woodhill

It also would have been obvious to modify Browne to include a length value, such as that proposed in Woodhill.[18] Dr. Clark explains that such a modification would have been desirable to further reduce the likelihood of hash collisions, and

[18] Alternatively Kantor also has a length value in its identifier and could be used.

would constitute the application of a known technique to a known device, ready for improvement, to yield predictable results. (Clark Decl. at ¶¶ 37-38; Ex. 1009.)

B. Grounds of Invalidity for Challenged Claims 1-4, 29-33 and 41 based on Langer as a Primary Reference

Ground 5: Langer Anticipates Challenged Claims 1-4, 29-33 and 41

Langer was not cited to the USPTO and not considered by the examiner during prosecution of the '791 patent. It is prior art under at least 35 U.S.C. § 102(b) and anticipates each of claims 1–4, 29–33 and 41 of the '791 patent.

Langer addresses the problem of distributing files over the Internet. Langer predates the advent of the World Wide Web, and therefore focuses on earlier file distribution technologies notably the *File Transfer Protocol* (FTP) and the Archie and WAIS search engines. (Langer at 2; Ex. 1003.) Langer provided his contribution to the "alt.sources" Usenet newsgroup. At the time, Usenet was one of the most effective channels for researchers to discuss current technical issue and distribute research materials.

Like the '791 patent and Browne, Langer recognizes the limitations inherent in the use of context- or location-based file identifiers, and the benefits of "uniquely identifying files which may have different names and/or be in different directories on different systems." (Langer at 3; Ex. 1003.) For example, identifiers that are tied to a physical server do not allow a user to select a site that is physically closer. (Langer at 3; Ex. 1003.)

Langer's solution is exactly the same as the '791 patent: determine a substantially unique identifier for each file based on the ***content*** of the file rather than its ***location***, and associate that file with the unique identifier (Langer at 3–4; Ex. 1003; *compare* '791 patent, col. 3, ll. 33-35; Ex. 1001.) Langer expressly recognizes that such an identifier may be calculated with an MD5 hash function:

> A simple method of defining a unique identifier that does NOT include a particular site identifier would be to use a hash function on the entire contents of the file. . . . I would suggest using a cryptographic hash function such as MD5 which generates a 16 byte result.

(Langer at 4; Ex. 1003.) The'791 patent tracks Langer's solution (which predates it by almost four years) down to the choice of the same MD5 hash function.

Langer's unique identifier depends on ***all,*** and ***only*** the contents of the data item. Langer expressly mentions computing a hash function on "the ***entire*** contents of the file." (Langer at 4 (emphasis added); Ex. 1003.) The MD5 algorithm receives ***only*** one input, namely, the contents of the file being hashed. As a consequence, if two files have the same contents, they will necessarily have the same MD5 hash, even if they have different names, or are stored at different locations on the Internet.

Like Browne, Langer also specifically addresses compound data items, such as archived files that are part of the same package. (Langer at 5; Ex. 1003.) Once again, this is the same algorithm adopted years later by the '791 patent to compute

True Names for "compound data items." ('791 patent, col. 14, ll. 1–27 and Fig. 10(b); Ex. 1001.)

Langer uses these substantially unique identifiers with a central database server, such as the Archie and WAIS search engines, that associates MD5 hashes with physical locations. (Langer at 3–4; Ex. 1003.) Assimilating new content to Langer's system of unique identifiers is straightforward. When an FTP server adds a file to its repository, it notifies the central database server of the new pairing between the file's MD5 hash and its location. (Langer at 4; Ex. 1003.)

Based on this infrastructure, Langer allows a client computer to determine whether a file is present in the system using its substantially unique identifier, and to access the file using this identifier. For example, a user can query the Archie or WAIS search engines to find which FTP server holds a copy of a file with a specified MD5 hash. (Langer at 3–4; Ex. 1003; *see also, e.g.*, EARN Staff, "Guide to Network Resource Tools," Internet RFC 1580 (March 1994) at 23–26 (WAIS), 29–31, 36–37 (Archie) ; Ex. 1034.) If a user tries to search for an MD5 hash that is not in the database, the search engine will not identify any servers, thus providing an immediate determination of whether a particular file is present in the system. If the file is present in the system, the client computer can access the file from one of

the previously-identified file servers, again using the file's unique identifier.[19] Langer's central database server thus directly anticipates the True File Registry of the '791 patent, which provides a list of the locations, such as file servers, where a file with a given True Name is stored. ('791 patent, col. 34, ll. 17–19; Ex. 1001.)

Langer also allows a client computer to determine whether a file is present at a particular location in the system, and to provide the file from another location if it is not present locally. Langer addresses the specific problems of dial-up connections by the use of intermediate cache servers:

> For dial-up sites a mail-server request could be chained until it reached a site with directory access, and the files requested added to temporary caches on the way back.

(Langer at 4–5; Ex. 1003.) Langer is describing a chain of servers, stretching from the source of the request (the client) to a server that holds a copy of the requested file. Each of the "temporary caches" along the chain first determines if it holds a copy of the requested file; if not, it forwards the request to the next cache in the

---

[19] Similarly to Browne, Langer proposes to alias MD5 signatures to actual file names on the server: "A simple ftp implementation would just hardlink every file available for ftp to a filename encoding of it's [sic] MD5 token. Users would then ftp the directory path and filename of the MD5 token and obtain the file." (Langer at 4, Ex. 1003.)

chain, until it reaches a site that can satisfy the request.[20] Consequently, if a particular cache determines that it holds the file, the particular location holding the file is by necessity also determined.

As set forth in detail in the attached claim chart (Exhibit 1039), and as confirmed by Dr. Clark (Clark Decl., ¶¶ 39-55), Langer anticipates each of claims 1–4, 29–33 and 41 of the '791 patent. For example, for claim 1, quoted above, with regard to the "identity means," Dr. Clark confirms that Langer meets this element because it determines the MD5 hash of the contents of each file. (Clark Decl, ¶¶ 41-43; Ex. 1009, Langer at 4, 5, Ex. 1003.) The MD5 hash is determined based on ***all*** and ***only*** the contents of the file, and that two identical files will always have the same identifier. (Clark Decl., ¶ 41-43; Ex. 1009, Langer at 3-6, Ex. 1003.)

With regard to the "existence means," Dr. Clark confirms that Langer meets this element as well, because it discloses using the Archie or WAIS ssearch engines to determine a location where a file with a particular MD5 hash is stored.

---

[20] Dr. Clark confirms that a cache is interposed between a source of data and a user of data. (Clark Decl. at ¶ 45.) In this case, one or more temporary caches are interposed between a server (source) and a client (user). (*Id.*) When the client sends a request to the source, each of the caches intercepts the request, returns the requested data if it can, and otherwise forwards the request to the next cache in the chain. (*Id.*)

(Clark Decl., ¶ 44-47; Ex. 1009, Langer at 4, 5, Ex. 1003.) If no locations are returned, the file is not stored anywhere in the system. (Clark Decl., ¶45 ; Ex. 1009, Langer at 4, Ex. 1003.) The search engines receive a search term (MD5 hash) and find the corresponding location(s) by examining the MD5 hashes in the database, and comparing them to the search term. (Clark Decl., ¶ 45-47; Ex. 1009, Langer at 4, 5, Ex. 1003.)

Similarly, for claim 30, Dr. Clark confirms that Langer determines a substantially unique identifier for a data item, and accesses the data item using the identifier, for essentially the same reasons. (Clark Decl., ¶¶ 56.) . Dr. Clark identifies the locations of the limitations of the claim in the reference, for example "a method of identifying a data item. . . comprising:" (Clark Decl. ¶¶ 39-43, Langer at 4, 5 ; Ex. 1003.) , "determining a . . . same identifier; and" (Clark Decl. ¶¶ 41-43, Langer at 4, 5; Ex. 1003.) and "accessing . . . of the data item" (Clark Decl. ¶¶ 54, 55, Langer at 3,4; Ex. 1003.)

A similar analysis reapplies to the remaining claims, as supported by Dr. Clark (Ex. 1009) and the claims chart (Ex. 1039). Langer discloses, for example, a "local existence means" (claim 2 and 3) at 4, 5; "each location contains . . . at said particular location in the system" (claim 3) at 4; a "data associating means" and an "access means" (claim 4) at 3, 4; "at least a file, database. . . object class" (claim 29) at 3, 4; "making and maintaining. . . via the association" (claim 31) at 3, 4, 5;

"assimilating a new data item. . .with its identifier" (claim 32) at 4, 5; "for a given data identifier . . . to the current location" (claim 41) at 3, 4; and "based on the determining. . . present at the destination" (claim 33) at 3, 4. The remaining limitations of claim 33 are similar to those in claim 30, and the discussion, *supra*, is incorporated by reference.

Ground 6: Challenged Claims 1-4 and 29 are Unpatentable as Obvious in view of Langer in combination with Woodhill

To the extent PersonalWeb contends that Langer does not meet the claim limitation of "identity means . . ." because Langer does not add the data item's length to the identifier, it would have been obvious to modify Langer to include a length value, such as that proposed in Woodhill.[21] (Woodhill at col. 4, ll. 43-46; Ex. 1005.) Dr. Clark confirms that such a modification would have been desirable to reduce the likelihood of hash collisions even further. (Clark Decl. at ¶¶ 57-58; Ex. 1009, Claim Chart; Ex. 1039.)

C. Grounds of Invalidity for Challenged Claims 1-4, 29-33 and 41 based on Kantor as a Primary Reference

Ground 7: Kantor Anticipates Challenged Claims 1-3, 29, and 33

Kantor was not cited to the USPTO and not considered during prosecution of the '791 patent. It is prior art under at least 35 U.S.C. § 102(b) and anticipates each of claims 1-3, 29, and 33 of the '791 patent.

---

[21] Alternatively Kantor also has a length value in its identifier and could be used.

Kantor is a published manual that describes a software program called the Frederick W. Kantor Contents-Signature System Version 1.22 ("FWKCS"). (Kantor at Title Page; Ex. 1004.) Like the '791 patent, Kantor addresses the shortcomings of context- or location-dependent file identifiers. (Kantor at Preface 1; Ex. 1004; *compare* '791 patent, col. 3, ll. 6-20; Ex. 1001.) These include the "problem of duplicate files on electronic bulletin board systems" or BBSs.[22] (*Id.*; Ex. 1004.) BBS users would unwittingly or intentionally upload files to a bulletin board, which the bulletin board already had. (*Id.* ; Ex. 1004.) Consequently, bulletin board operators "were paying for hardware to provide capacity for these spurious [duplicate] files, and spending many hours trying to find and delete them." (*Id.*; Ex. 1004) FWKCS was used by notable BBSs, such as "The Invention Factory," to detect and remove these duplicate files. (*Id.* at 5; Ex. 1004.)

Kantor uses the same solution that would be later proposed in the '791 patent: associating a unique identifier with the ***contents*** of a file, rather than with the ***location*** of the file. Kantor calls these identifiers "contents-signatures," and uses them to uniquely identify files based on their contents, not their name or location. (*Id.*; Ex. 1004; *compare* '791 patent, col. 3, ll. 6-20; Ex. 1001) These signatures

---

[22] Before the World Wide Web, computers "dialed into" a file server or network of servers where users could exchange files or other information by uploading or downloading files.

could be used, for example, to detect and delete existing duplicate content stored on the BBS system. (*e.g., id.* at 2-3; Ex. 1004) They also could be used to avoid storing unwanted duplicates, by automatically processing subsequent file uploads to create a corresponding signature to detect whether or not those files were already stored on the system. (*Id.* at 5.)

FWKCS computes the contents-signature based on ***all*** of the data in a file, and ***only*** the data in the file. (*See id.* at 6-8.) Specifically, the contents signature is constructed with "the 32-bit CRC [cyclic redundancy check][23] of the file contents and the uncompressed file-length." (*Id.*; Ex. 1004.) The CRC and the file length (*i.e.,* file size) both depend on all of, and only, the contents of a file. Consequently, FWKCS computes each contents-signature using all of, and only, the data in the file. Therefore, two files with the same content necessarily have the same contents-signature. (*See id.*; Ex. 1004.) Each contents-signature is location-independent and in no way depends on the file's pathname or context. Moreover, this contents-signature, comprising a CRC hash with an appended length, is exactly like the technique described in the '791 patent. That is, just as Kantor creates a contents-signature with a CRC hash and a length value, the '791 patent computes

[23] As Dr. Clark explains a CRC is a commonly used hash function that processes each and every byte of the input file. (Clark Decl., ¶ 62; Ex. 1009.)

an identifier with a MD hash and a length value. (*Id.* at 6-8; Ex. 1004; *compare* '791 patent, Fig. 10A and col. 14, ll. 1-12; Ex. 1001.)

The contents-signature identifiers uniquely identify the files in a system. As Kantor explains, each contents-signature is "a string of bits generated from the contents of a file, long enough and suitably generated so as to provide some desirably low probability that two different files would give rise to the identical string of bits." (*Id.* at 6.) Thus, like the '791 patent, FWKCS contents-signatures are "substantially unique."

Kantor also describes a file identifier for compound data items just like the identifier in the '791 patent. (See '791 patent, col. 14 ll. 12-31 and Fig. 10B; Ex. 1001.) Kantor's "zipfile contents signatures" – like Browne's and Langer's identifiers for compound data items – depend on all of the data and only the data in the data item and track the algorithm in the '791 patent. (*Id.*; Ex. 1004.)

FWKCS provides many operations for working with these contents-signatures. Like the '791 patent, these operations include means for determining whether a file is present on a server based on the file's substantially unique identifier. For example, FWKCS computes the contents-signatures for all of the files in the system and stores them in a master list called CSLIST.SRT, similar to the "True Name Registry" of the '791 patent. (Id. at 18; Ex. 1004; compare '791 patent, col. 34, ll. 17-19; Ex. 1001). After creating the master list, FWKCS may

compare the contents-signatures of each file to detect duplicates. (Id. at 50; Ex. 1004.) FWKCS logs the detected duplicates, and system operators can configure FWKCS to delete them automatically. (Id. at 101; Ex. 1004.) Another operation prevents the BBS from storing new files that would be duplicates of files already on the system. In this case, FWKCS computes the contents-signature of an uploaded file and uses that contents-signature to check whether the uploaded file is already present on the system. (Id. at 5; Ex. 1004.) A third operation, called "Lookup," allows a BBS user to compute the contents-signature of a file and send it to the BBS so it can compare that signature to the master list CSLIST.SRT to determine if the file is already present on the system. (*Id.* at 96-97; Ex. 1004.) This Lookup feature also works like the '791 embodiment, and "helps you avoid uploading duplicate or redundant material" because "a person can ask ahead to find out if material which he/she is thinking of uploading is already on a BBS." (Kantor at 173, 96; Ex. 1004; *compare* '791 patent, col. 33, ll. 28-33; Ex. 1001.)

As set forth in detail in the attached claim chart (Exhibit 1040), and as confirmed by Dr. Clark (Clark Decl. at ¶¶ 59-73; Ex. 1009), Kantor anticipates each of claims 1-3, 29 and 33 of the '791 patent. For example, for claim 1, quoted above, with regard to the "identity means," Dr. Clark confirms that Kantor meets this element because it calculates a contents-signature as a "substantially unique identifier" that is determined using "all" of the data in the file and "only" the data in the

file. (Clark Decl. at ¶ 61-63; Ex. 1009, Kantor 10, 11, Ex. 1004.) Dr. Clark also confirms that Kantor's use of the CRC and size of the file's contents will result in two identical files always having the same contents-signature. (Clark Decl. at ¶ 62; Ex. 1009, Kantor 6, 8, Ex. 1004.) In fact, this is the very same technique as the '791 patent. (Clark Decl. at ¶ 69; Ex. 1009.)

With regard to the "existence means," Dr. Clark confirms that Kantor teaches this element as well. (Clark Decl. at ¶63-69; Ex. 1009, Kantor 18, 45, Ex. 1004.) For example, FWKCS allows a user to query a bulletin board system (BBS) with a contents-signature to determine where the corresponding file is stored in the BBS. (Clark Decl. at ¶ 66; Ex. 1009, Kantor 97, 173; Ex. 1004.) After receiving such a query, the BBS examines the contents-signatures (identifier), and determines whether the file is stored on the BBS and, if so, where (i.e., to "determin[e] whether a particular data item [i.e., file] is present in the system"). (Clark Decl. at ¶ 66; Ex. 1009; Kantor 97, 173, Ex. 1004) More specifically, it does so by examining the CSLIST.SRT or master file that stores contents-signatures for all of the files stored on the BBS. (Clark Decl. at ¶¶65-66; Ex. 1009; Kantor 18, 45, Ex. 1004.) Again, this is the very technique used in the '791 patent (*i.e.*, the True File Registry). (Clark Decl. at ¶ 69; Ex. 1009.)

Similarly, for claim 30, Dr. Clark confirms that Kantor determines a substantially unique identifier for a data item, and accesses the data item using the

identifier, for essentially the same reasons. (Clark Decl., ¶¶ 74.) Dr. Clark identifies the locations of the limitations of the claim in the reference, for example "a method of identifying a data item. . . comprising:" (Clark Decl. at ¶¶ 61-63, 74-76; Ex. 1009, Kantor at 8, 10, 11, 51; Ex. 1004.) , "determining a . . . same identifier; and" (Clark Decl. at ¶¶ 61-63; Ex. 1009, Kantor at 8, 10, 11, 51; Ex. 1004) and "accessing . . . of the data item" (Clark Decl. at ¶ 76-78; Ex. 1009, Kantor at 113, 121, 124; Ex. 1004.)

A similar analysis applies to the claims 2, 3, 29, and 33 as supported by Dr. Clark (Ex. 1009) and the claims chart (Ex. 1040). As Dr. Clark confirms, Kantor discloses, for example, a "local existence means" (claims 2 and 3) at p. 148; "each location contains . . . at said particular location in the system" (claim 3) at p. 148; "at least a file, database. . . object class" (claim 29) at p. 3; and "based on the determining. . . present at the destination" (claim 33) at p. 81.

Ground 8: Challenged Claims 4, 30-32 and 41 are Unpatentable as Obvious in view of Kantor

In the event that PersonalWeb contends that Kantor does not meet the "access means," or "accessing a data item in the system using the identifier of the data item" limitations, because BBS clients typically connected to a BBS and requested files based on their name and location, a person of ordinary skill in the art would have found it obvious to modify the BBS commands, including download or read commands, to permit identifying files based on Kantor's contents-signatures or

zipfile contents-signatures. (Clark Decl. at ¶¶ 75-79; Ex. 1009.) Among other things, this would facilitate integrity checking by more precisely specifying the file of interest by its content, and improve accuracy. (Clark Decl. at ¶ 79; Ex. 1009.) Kantor shows that such a modification would be easy to implement. For example, Kantor already utilized contents-signatures as parameters specified in certain user commands, such as the "Lookup" operation (*see* Kantor at 97 and 173; Ex. 1004), and it would have been straightforward to similarly allow download and read commands to identify a file based on a contents-signature. (Clark Decl. at ¶¶ 75-79; Ex. 1009.) Moreover, it would be an easy matter for the user to obtain the contents-signatures for the files of interest. For example, the signatures could be shared among users. In addition, the signatures could be provided to the user by the BBS itself using the Precheck operation or an easily modified version of it. (Clark Decl. at ¶¶ 67, 75-79; Ex. 1009.)

Ground 9: Challenged Claims 1-4 and 29 are Unpatentable as Obvious in view of Kantor in view of Langer

In the event PersonalWeb contends that Kantor does not meet the "identity means" limitation because Kantor's contents-signature uses a CRC hash plus file length, rather than the MD5 hash plus file length disclosed in the '791 patent, a person of ordinary skill in the art would have found it obvious to modify Kantor in view of Langer to meet that limitation. As Dr. Clark confirms, Kantor expressly discusses a statistical experimentation that he conducted to prove that the contents-

signature format would be a substantially unique identifier on bulletin board systems. (Clark Decl. at ¶ 80; Ex. 1009; *see also* Kantor at pp. 10-11 ("Based on these experimental results, the enhanced accuracy provided by the FWKCS contents-signature appears to have resulted, in effect, in a typical pairwise statistical error rate of less than one part in ten trillion."); Ex. 1004.) Langer discloses using an MD5 hash of a file's contents as part of the file's identifier. (Langer at 4-6; Ex. 1003.) As Dr. Clark confirms, modifying Kantor's contents-signature to use an MD5 hash instead of CRC hash would have been desirable to improve the statistical error rate of FWKCS using a hash function widely accepted in the community. (*Id.;* Ex. 1009; *see also* Langer at 4-6; Ex. 1003.) This modification would constitute the application of a known technique to a known device, ready for improvement, to yield predictable results, and therefore it would be obvious to a person of ordinary skill in the art. (*Id.*)

D. Grounds of Invalidity for Challenged Claims 1-4, 29-33 and 41 based on Woodhill as a Primary Reference

Ground 10: Woodhill Anticipates Challenged Claims 1-4, 29-33 and 41

Woodhill was not cited to the USPTO and not considered by the examiner during prosecution of the '791 patent. It is prior art under at least 35 U.S.C. § 102(e) and anticipates each of claims 1–4, 29–33 and 41 of the '791 patent.

Woodhill discloses a distributed storage management system, with mechanisms for backing up and later restoring (i.e., accessing) the files stored by each

computer of the system. (*Woodhill* at col. 2, ll. 39-49; Ex. 1005.) Figure 1 of Woodhill shows some of the basic elements of the system:

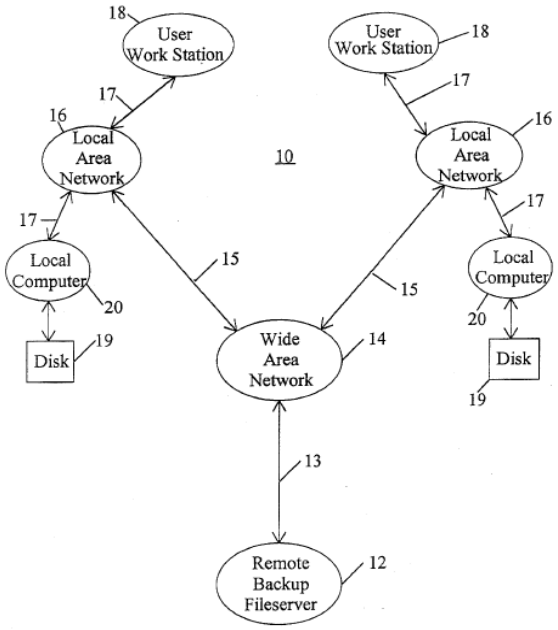


FIG. 1

To backup files or later access those files, Woodhill breaks the files into "binary objects," having a size of one megabyte or less. (*Id*. at col. 4, ll. 12-30; Ex. 1005.) These files, and their component binary objects, are distributed across local computers 20 connected to local area networks 16. (*Id*. at col. 3, ll. 24-44; Ex. 1005.) These local computers 20 and their local area networks 16, in turn, are in communication with a remote backup file server 12 through a wide area network 14. (*Id*. at col. 3, ll. 12-27; Ex. 1005.) For each backup cycle, only new or changed files are provided to these backup sites. (*Id*. at col. 9, ll. 36-38; Ex. 1005.) Once files are backed up, the system allows a local computer 20 to access them

from a different local computer 20 or from the remote backup file server 12, as needed. (*Id*. at col. 9, ll. 39-44; Ex. 1005.)

To perform a backup, each local computer 20 executes the Distributed Storage Manager program. (*Id*. at col. 5, ll. 3-20; Ex. 1005.) This program first identifies all of the files that must be processed during the present backup cycle. (*Id*. at col. 5, ll. 13-20; Ex. 1005.) This set of files includes previously backed up files, as well as new files identified by "scan[ing] all disk drives 19 on the local computer 20 that are to be backed up." (*Id*. at col. 5, ll. 46-49; Ex. 1005.) Once the set of files to be backed up is identified, these files are subdivided into one or more binary objects. (*Id*. at col. 7, ll. 40-55; Ex. 1005.)

To identify and later compare binary objects, Woodhill determines a Binary Object Identification Record, including a Binary Object Identifier, for each binary object. (*Id*. at col. 7, l. 60 – col. 8, l. 1; Ex. 1005.) Each "Binary Object Identifier is calculated from the contents of the data" of the corresponding binary object (*Id*. at col. 8, ll. 40-41; Ex. 1005). As Woodhill makes clear, the "***critical feature to be recognized in creating a Binary Object Identifier 74 is that the identifier should be based on the contents of the binary object*** so that the Binary Object Identifier 74 changes when the contents of the binary object changes." (*Id*. at col. 8, ll. 58-62; Ex. 1005.) Accordingly, "the possibility of two different binary objects being assigned the same Binary Object Identifier 74 [is] very small," and each Binary

Object Identifier 74 " uniquely identif[ies] a particular binary object." (*Id*. at col. 8, ll. 33-36; Ex. 1005.)

Figure 3 of Woodhill shows the Binary Object Identifier portion of a Binary Object Identification Record. This Binary Object Identifier is 128 bits long and includes Binary Object Size, a Binary Object Cyclical Redundancy Check (CRC), a Binary Object Longitudinal Redundancy Check (LRC), and a Binary Object Hash. (*Id*. at col. 7, l. 64 – col. 8, l. 4; Ex. 1005.) Binary Object Size is determined by adding up the number of bytes of the binary object. (*Id*. at col. 8, ll. 4-5; Ex. 1005) Binary Object CRC is "calculated against the contents of the binary object taken one (1) byte (8 bits) at a time." (*Id*. at col. 8, ll. 5-9; Ex. 1005.) Binary Object LRC is "calculated against the contents of the binary object taken four (4) bytes (32 bits) at a time." (*Id*. at col. 8, ll. 11-20; Ex. 1005.) Finally, Binary Object Hash 70 is "calculated against the contents of the binary object taken one (1) word (16 bits) at a time":

```
HASH = (initialized value)
for each word (16 bits) of the binary object:
   rotate current HASH value by 5 bits
   HASH = HASH + 1
   HASH = HASH + (current word (16 bits) of binary object)
end loop
```

(*Id*. at col. 8, ll. 21-31; Ex. 1005.) Each of the fields is accordingly "calculated from the contents of each binary object." (*Id.* at col. 8, lines 1- 33; Ex. 1005.)

Binary Object Identifiers 74, like the True Names of the '791 patent, are thus based on the contents of the binary objects. (*Cf.* '791 patent, Figure 10A (calculating True Name based on MD5 hash of data item and length of data item); Ex. 1001.) Each of the Binary Object CRC, Binary Object LRC, and Binary Object Hash functions cycle through every bit of the binary object and incorporates that bit when determining its resulting value. (Woodhill at col. 7, l. 64 – col. 8, l. 31; Ex. 1005.) Thus, the Binary Object Identifier is determined using all of the data of a binary object (i.e., each and every bit). Furthermore, only data from the binary object is used to determine the identifier, not data "from an external and arbitrary source." (*Id*. at col. 8, ll. 38-42; Ex. 1005.)

By determining content-based identifiers for binary objects in this way, "duplicate binary objects, even if resident on different types of computers in a heterogeneous network, can be recognized from their identical Binary Object Identifiers 74." (*Id*. at col. 8, ll. 62-62; Ex. 1005.) Thus, the backup and restore system implemented by Woodhill can determine the existence of duplicate binary objects across the network using their content-based identifiers.

Woodhill uses the Binary Object Identifiers, for example, to achieve efficient backup operations by "determin[ing] which parts of a file have changed and only back[ing] up the changed data instead of backing up all of the data associated with a file when only a small portion of the file has been modified." (*Id*. at col. 9,

ll. 14-27; Ex. 1005.) In other words, "only those binary objects associated with the file that have changed must be backed up." (*Id*. at col. 9, ll. 7-9; Ex. 1005.) This process works by "comparing the current value of the binary object identifier associated with a particular binary object to one or more previous values of the binary object identifier associated with that particular binary object; and . . . selectively copying binary objects in response to the step for comparing." (*Id*. at col. 2, ll. 33-38; Ex. 1005.)

Woodhill also uses the identifiers for efficient file access, for example, if a backed-up binary object is stored "somewhere on the local area network 16 other than on the local computer 20 on which it normally resides" and also on a remote backup file server. (*Id*. at col. 9, ll. 31-38; Ex. 1005.) "[T]he local copies of binary objects serve to handle very fast restores for most restore requests that occur on the local area network. If the local backup copy of a file does not exist or a prior version of a file is required, it must be restored from the remote backup file server 12." (*Id*. at col. 10, ll. 30-34; Ex. 1005.)

As set forth in detail in the attached claim chart (Exhibit 1041), and as confirmed by Dr. Clark (Clark Decl., ¶¶81-95; Ex. 1009), Woodhill anticipates each of claims 1–4, 29–33, and 41 of the '791 patent. For example, for claim 1, quoted above, with regard to the "identity means," Dr. Clark confirms that Woodhill meets this element because the Binary Object Identifiers are "substantially unique identi-

fiers" determined using the contents in the binary objects. (Clark Decl. at ¶¶ 83-84; Ex. 1009, Woodhill col. 7, line 60 – col. 8, line 1, Fig. 3; Ex. 1005.) In calculating the identifiers, Woodhill uses "all of" the data of a binary object and "only" the data of a binary object. (Clark Decl. at ¶ 84; Ex. 1009,Woodhill col. 8, ll. 1-31; Ex. 1005.) Because each Binary Object Identifier is based on the data of a binary object, two identical binary objects in Woodhill's system will have the same Binary Object Identifier. (Clark Decl. at ¶ 84; Ex. 1009, Woodhill col. 8, ll. 1-31; Ex. 1005.)

With regard to the "existence means," Dr. Clark confirms that the existence of particular binary objects, both on any local computer 20 part of a local area network 16, and on a remote backup file server 24, is determined by examining those Binary Object Identifiers. (Clark Decl. at ¶ 85-87; Ex. 1009, Woodhill col. 8, l. 62-col. 9, l. 23; Ex. 1005.) Woodhill does so, for example, by checking whether a binary object has changed since it was last backed up using its Binary Object Identifier, and by checking whether a local copy of a binary object is available to restore using its Binary Object Identifier. (Clark Decl. at ¶ 86; Ex. 1009, Woodhill col. 9, ll. 14-22; Ex. 1005.)

Similarly, for claim 30, Dr. Clark confirms that Woodhill determines a substantially unique identifier for a data item, and accesses the data item using the identifier, for essentially the same reasons. (Clark Decl., ¶¶ 96.) Dr. Clark identi-

fies the locations of the limitations of the claim in the reference, for example "a method of identifying a data item. . . comprising:" (Clark Decl. at ¶ 84; Ex. 1009, Woodhill at col. 7, l. 60 – col. 8, line 31; Ex. 1005.) , "determining a . . . same identifier; and" (Clark Decl. at ¶¶ 80-84; Ex. 1009, Woodhill at col. 7, l. 60 – col. 8, l. 31; Ex. 1005.) and "accessing . . . of the data item" (Clark Decl. at ¶ 94, 95; Ex. 1009, Woodhill at col. 18, ll 11-23, col. 11, ll. 12-15, Figs. 5A and 5J; Ex. 1005.)

A similar analysis reapplies to the remaining claims, as supported by Dr. Clark (Ex. 1009) and the claims chart (Ex. 1041). As Dr. Clark confirms, Woodhill discloses, for example, a "local existence means" (claim 2 and 3) at col. 9, lines 5-23; "each location contains . . . at said particular location in the system" (claim 3) at col. 9, lines 5-23; a "data associating means" and an "access means" (claim 4) at col. 7, line 60 – col. 8, line 65 and col. 18, lines 11-23; "at least a file, database. . . object class" (claim 29) at col. 7 lines 51-55 ; "making and maintaining. . . via the association" (claim 31) at col. 7, line 60 – col. 8, line 65 and col. 18, lines 11-23; "assimilating a new data item. . .with its identifier" (claim 32) at col. 7, line 60 – col. 8, line 65; "for a given data identifier . . . to the current location" (claim 41) at col. 9, lines 5-23; and "based on the determining. . . present at the destination" (claim 33) at col. 9, lines 5-23. The remaining limitations of claim 33

are similar to those in claim 30, and the discussion, *supra*, is incorporated by reference.

Ground 11: Challenged Claims 1-4 and 29 are Unpatentable as Obvious in view of Woodhill

In the event that PersonalWeb contends that Woodhill does not meet the "identity means" limitation because it does not compute an MD5 hash value, a person of ordinary skill in the art would have found it obvious to modify Woodhill to include such a calculation. (Clark Decl. at ¶ 97-98; Ex. 1009.) For example, Woodhill states that "[t]hose of ordinary skill in the art will recognize that there exist many different ways of establishing the Binary Object Identifier 74 (e.g., establishing a Binary Object Identifier 74 of a different length or utilizing different calculations) and that the procedure set forth above is only one way of establishing the Binary Object Identifier 74." (Woodhill at col. 8, ll. 52-58; Ex. 1005). As Dr. Clark confirms, modifying Woodhill's Binary Object Identifiers to use an MD5 hash function would have been one such different calculation. (*Id.*) This modification would constitute a simple substitution of one known element for another to obtain predictable results, and therefore it would be obvious to a person of ordinary skill in the art exercising ordinary creativity. (*Id.*)

Ground 12: Challenged Claims 1-4 and 29 are Unpatentable as Obvious in view of Woodhill in view of Kantor

Furthermore, in the event that PersonalWeb contends that the "identity means" requires computing data item identifiers by partitioning data items into segments, and computing segment identifiers using these segments, (*see* '791 patent, Fig. 10B; Ex. 1001), a person of ordinary skill exercising ordinary creativity would have found it obvious to combine Woodhill with Kantor's computation of a "zipfile contents signature." (Clark Decl., ¶¶ 99-100; Ex. 1009.) As Dr. Clark explains, a person of ordinary skill in the art exercising ordinary creativity would have been motivated to modify the Binary Object Identifier calculation to use the technique disclosed by Kantor. (*Id.*) Modification of the Binary Object Identifier in this way would constitute a simple substitution of one known element for another to obtain predictable results, and therefore it would be obvious to a person of ordinary skill in the art. (*Id.*)

VII. CONCLUSION

Based on the foregoing, it is clear that claims 1-4, 29-33 and 41 of the '791 Patent recite subject matter that is either anticipated or obvious. The Petitioner requests institution of an *inter partes* review and cancelation of those claims.

Respectfully Submitted,

/David L. Cavanaugh/

David L. Cavanaugh
Registration No. 36,476

CERTIFICATE OF SERVICE

I hereby certify that, on December 15, 2012, I caused a true and correct copy of the foregoing materials:

- Petition for *Inter Partes Review* of U.S. Patent No. 5,978,791
- Exhibits 1001-1041
- Fee Summary Page
- EMC Corp. Power of Attorney
- VMware, Inc. Power of Attorney

to be served via Express Mail on the following attorney of record as listed on PAIR:

Pillsbury Winthrop Shaw Pittman, LLP

PO Box 10500

McLean, Virginia 22102

/David L. Cavanaugh/

David L. Cavanaugh

Registration No. 36,476

Table of Exhibits for U. S. Patent 5,978,791 Petition for *Inter Partes* Review

Exhibit	Description
1001.	U.S. Patent No. 5,978,791
1002.	S. Browne et al., "Location-Independent Naming for Virtual Distributed Software Repositories," University of Tennessee Technical Report CS-95-278 (Feb. 1995)
1003.	A. Langer, "Re: dl/describe (File descriptions)," post to the "alt.sources" newsgroup on August 7, 1991
1004.	F. W. Kantor, "The Frederick W. Kantor Contents-Signature System Version 1.22," FWKCS122.REF (August 10, 1993)
1005.	Woodhill et al., U.S. Patent No. 5,649,196, entitled "System and Method For Distributed Storage Management on Networked Computer Systems Using Binary Object Identifiers."
1006.	S. Browne et al., "Location-Independent Naming for Virtual Distributed Software Repositories," http://www.netlib.org/utk/papers/lifn/main.html (Nov. 11, 1994)
1007.	K. Moore et al., "An Architecture for Bulk File Distribution," Network Working Group Internet Draft (July 27, 1994)
1008.	Chart of Patent Family Members
1009.	Declaration of Dr. Douglas W. Clark PH.D
1010.	Banisar et al., The Third CPSR Cryptography and Privacy Conference at 509 (1993)
1011.	G. D. Knott, Hashing functions, The Computer Journal 18 (1975), no. 3, p.265.
1012.	R. Rivest, "The MD5 Message-Digest Algorithm," Internet RFC 1321 (Apr. 1992)
1013.	McGraw-Hill Dictionary of Scientific and Technical Terms, (4th ed., 1989)

1014.	B. Kaliski, "A Survey of Encryption Standards, " IEEE Micro (Dec. 1993)
1015.	Rabin, Fingerprinting by Random Polynomials, Center for Research in Computing Technology, Harvard University, Report TR-15-81
1016.	U.Manber, "Finding Similar Files in a Larger File System", University of Arizonia Technical Report (1994)
1017.	D.R. McGregor and J.A. Mariani 'Fingerprinting' – A Technique for File Identification and Maintenance, Software practice & Experience 1165 (1982)
1018.	T. Berners-Lee et al., "Uniform Resource Locators (URL)," Internet RFC 1738 (Dec. 1994)
1019.	U. S. Patent 6,415, 280 Prosecution History, Response (August 22, 2001)
1020.	EP Pub. No. EP0826181A1 Prosecution History, Annex to the communication dated May 8, 2009
1021.	EP Pub. No. EP0826181A1 Prosecution History, Reply to communication from the Examining Division dated November 18, 2009
1022.	EP Pub. No. EP0826181A1 Prosecution History, Annex to the communication dated March 14, 2012
1023.	EP Pub. No. EP0826181A1 Prosecution History, Closing of Application dated June 14, 2012
1024.	U.S. Patent 5,978,791 Prosecution History, Application as filed on April 11, 1995
1025.	U.S. Patent 5,978,791 Prosecution History, Office Action of September 7, 1996
1026.	U.S. Patent 5,978,791 Prosecution History, Amendment of March 12, 1997

1027.	U.S. Patent 5,978,791 Prosecution History, Office Action of May 30, 1997
1028.	U.S. Patent 5,978,791 Prosecution History, Amendment of August 29, 1997
1029.	E. Babb, "Implementing a Relational Database by Means of Specialized Hardware, ACM Transactions on Database Systems", Vol. 4, No.1, at 2-4, March 1979
1030.	D. Bitton and D. DeWitt, "Duplicate Record Elimination in Large Data Files, ACM Transactions on Database Systems", Vol. 8, No. 2, at 255 – 265 (June 1983)
1031.	R. Williams, "An algorithm for matching text (possibly original)", posted to the "comp.compression" newsgroup on January 27, 1992
1032.	R. Williams, "An Introduction to Digest Algorithms," Rocksoft (Nov. 1994)
1033.	P. Deutsch et al., "How to Use Anonymous FTP," Internet RFC 1635
1034.	EARN Staff, "Guide to Network Resource Tools," Internet RFC 1580 (March 1994)
1035.	C. Adie, "Network Access to Multimedia Information," Internet RFC 1614 (May 1994), at 29–34
1036.	G. Kessler et al., "A Primer On Internet and TCP/IP Tools," Internet RFC 1739 (Dec. 1994)
1037.	Order Regarding Claim Construction, Akamai Tech., Inc. v. Digital Island, Inc., No. 00-11851-RWZ (D. Mass. Nov. 8, 2011) ECF No. 137
1038.	Invalidity Claim Chart in view of LIFN ("Browne")
1039.	Invalidity Claim Chart in view of Langer
1040.	Invalidity Claim Chart in view of FWKCS Contents -- Signature

	System Version 1.22 ("Kantor")
1041.	Invalidity Claim Chart in view of Woodhill